

Lección 1: Bases de datos relacionales

Motores de bases de datos

Un motor de bases de datos es un software diseñado para gestionar, almacenar y recuperar datos de manera eficiente. Funciona como una capa intermedia entre las aplicaciones y los datos almacenados en memoria física (hardware). Su objetivo principal es proporcionar un entorno para la manipulación de datos, permitiendo a los usuarios realizar consultas, actualizaciones, eliminaciones y cambios a los datos almacenados.

Todo motor de base de datos tiene que cumplir con ciertos requisitos de fiabilidad para evitar que se pierda información, también tienen que cumplir con características de eficiencia para que las consultas se hagan en tiempos adecuados a pesar de que la cantidad de datos sea considerablemente grande (complejidad algorítmica). Dentro de las características que debe cumplir se cuenta con:



Eficiencia: Debe ser capaz de almacenar y recuperar datos de manera eficiente, minimizando el tiempo de respuesta de las consultas y transacciones.

Integridad de los datos: Debe garantizar la integridad de los datos, asegurando que la información almacenada sea precisa y consistente.

Seguridad: Debe proporcionar mecanismos robustos para controlar el acceso a los datos y proteger la información sensible de accesos no autorizados.

Confiabilidad: Debe ser confiable y resistente a fallas, garantizando que los datos estén disponibles y que no se pierdan en caso de errores o fallos del sistema.

Escalabilidad: Debe ser capaz de manejar grandes volúmenes de datos y soportar un número creciente de usuarios y aplicaciones sin degradación del rendimiento.

Mantenibilidad: Debe ser fácil de mantener y administrar, permitiendo realizar tareas de configuración, monitoreo y optimización de manera sencilla.

Compatibilidad: Debe ser compatible con estándares y protocolos de la industria, facilitando la interoperabilidad con otras aplicaciones y sistemas.

Transaccionalidad: Debe soportar transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) para garantizar la integridad y consistencia de los datos durante las operaciones de escritura. Acerca ACID, es un acrónimo que se refiere al conjunto de 4 propiedades clave que definen una transacción: Atomicidad, Consistencia, Aislamiento y Durabilidad. Si una operación de base de datos tiene estas propiedades ACID, se la puede denominar transacción ACID, y los sistemas de almacenamiento de datos que aplican estas operaciones se denominan sistemas transaccionales. Las transacciones ACID garantizan que cada lectura, escritura o modificación de una tabla tenga las siguientes propiedades:

Atomicidad: cada declaración en una transacción (para leer, escribir, actualizar o eliminar datos) se trata como una sola unidad. O se ejecuta toda la declaración o no se ejecuta nada. Esta propiedad evita que se produzcan pérdidas y daños en los datos si, por ejemplo, la fuente de datos de transmisión falla a mitad de la transmisión.



Coherencia: garantiza que las transacciones solo realicen cambios en las tablas de manera predefinida y predecible. La coherencia transaccional garantiza que la corrupción o los errores en sus datos no creen consecuencias no deseadas para la integridad de su tabla.

Aislamiento: cuando varios usuarios leen y escriben desde la misma tabla a la vez, el aislamiento de sus transacciones garantiza que las transacciones simultáneas no interfieran ni se afecten entre sí. Cada solicitud puede ocurrir como si estuvieran ocurriendo una por una, aunque en realidad estén ocurriendo simultáneamente.

Durabilidad: garantiza que los cambios en sus datos realizados mediante transacciones ejecutadas con éxito se guardarán, incluso en caso de falla del sistema.



Coherencia: garantiza que las transacciones solo realicen cambios en las tablas de manera predefinida y predecible. La coherencia transaccional garantiza que la corrupción o los errores en sus datos no creen consecuencias no deseadas para la integridad de su tabla.

Aislamiento: cuando varios usuarios leen y escriben desde la misma tabla a la vez, el aislamiento de sus transacciones garantiza que las transacciones simultáneas no interfieran ni se afecten entre sí. Cada solicitud puede ocurrir como si estuvieran ocurriendo una por una, aunque en realidad estén ocurriendo simultáneamente.

Durabilidad: garantiza que los cambios en sus datos realizados mediante transacciones ejecutadas con éxito se guardarán, incluso en caso de falla del sistema.



¿Cuál es la diferencia entre una base de datos y una hoja de cálculo?

Tanto las bases de datos como las hojas de cálculo (como Microsoft Excel) son modos cómodos de almacenar información. Las principales diferencias entre los dos son la forma en la que se almacenan y manipulan los datos, las personas que pueden acceder a los datos a la vez y cuantos datos pueden leerse en un instante de tiempo.

Las hojas de cálculo se diseñaron originalmente para un usuario o para un pequeño número de usuarios que no necesiten hacer una manipulación de datos complicada. Las bases de datos, por otro lado, están diseñadas para contener recopilaciones mucho más grandes de información organizada, a veces en cantidades masivas. Las bases de datos permiten que muchos usuarios accedan y consulten los datos de forma rápida y segura al mismo tiempo mediante una lógica y un lenguaje estructurado.

Tipos de bases de datos

Existen muchos tipos diferentes de bases de datos. La mejor base de datos para una organización específica depende de cómo pretenda la organización utilizar los datos y de los datos en si mismos.

Bases de datos relacionales

Las bases de datos relacionales se hicieron predominantes en la década de 1980. Los elementos de una base de datos relacional se organizan como un conjunto de tablas con columnas y filas. La tecnología de bases de datos relacionales proporciona la forma más eficiente y flexible de acceder a información estructurada. Debido a que se acceden con el lenguaje estándar de consultas se suelen manejar con adaptaciones del estándar SQL.

Bases de datos orientadas a objetos

La información de una base de datos orientada a objetos se representa en forma de objetos, como en la programación orientada a objetos.

Bases de datos distribuidas

Una base de datos distribuida consta de dos o más archivos que se encuentran en sitios diferentes. La base de datos puede almacenarse en varios ordenadores, ubicarse en la misma ubicación física o repartirse en diferentes redes.

Almacenes de datos

Un repositorio central de datos, un data warehouse es un tipo de base de datos diseñado específicamente para consultas y análisis rápidos.

Bases de datos NoSQL

Una base de datos NoSQL, o base de datos no relacional, permite almacenar y manipular datos no estructurados y semiestructurados (a diferencia de una base de datos relacional, que define cómo se deben componer todos los datos insertados en la base de datos). Las bases de datos NoSQL se hicieron populares a medida que las aplicaciones web se volvían más comunes y complejas.

Bases de datos orientadas a grafos

Una base de datos orientada a grafos almacena datos relacionados con entidades y las relaciones entre entidades.

Bases de datos OLTP

Una base de datos OLTP es una base de datos rápida y analítica diseñada para que muchos usuarios realicen un gran número de transacciones.

Bases de datos en la nube

Una base de datos en la nube es una recopilación de datos, estructurados o no estructurados, que reside en una plataforma de cloud computing privada, pública o híbrida. Existen dos tipos de modelos de bases de datos en la nube: el modelo tradicional y el de base de datos como servicios (database as a service, DBaaS). Con DBaaS, un proveedor de servicios realiza las tareas administrativas y el mantenimiento.

Lenguajes basados en SQL

SQL es un estándar de de consultas estructuradas. Este estándar define cómo deben manejarse y mantenerse los datos en un sistema de bases de datos relacionales o RDBMS. Está diseñado para datos estructurados e integra múltiples instrucciones. Los comandos de lenguaje de consulta estructurada (SQL) son palabras clave o instrucciones SQL específicas que los desarrolladores utilizan para manipular los datos almacenados en una base de datos relacional. Puede clasificar los comandos SQL de la siguiente manera.

Lenguaje de definición de datos

El lenguaje de definición de datos (DDL) se refiere a comandos SQL que diseñan la estructura de la base de datos. Los ingenieros de bases de datos utilizan DDL para crear y modificar objetos de bases de datos en función de las necesidades empresariales. Por ejemplo, el ingeniero de bases de datos utiliza el comando CREATE para crear objetos de base de datos, como tablas, vistas e índices.

Idioma de consulta de datos

Las instrucciones del lenguaje de manipulación de datos (DML) escriben información nueva o modifican los registros existentes en una base de datos relacional. Por ejemplo, una aplicación usa el comando INSERT para almacenar un nuevo registro en la base de datos.

Idioma de control de datos

Los administradores de bases de datos usan el lenguaje de control de datos (DCL) para administrar o autorizar el acceso a la base de datos. Por ejemplo, pueden usar el comando GRANT para permitir que ciertas aplicaciones manipulen una o más tablas.

Idioma de control de transacciones

El motor relacional utiliza el lenguaje de control de transacciones (TCL) para hacer cambios en la base de datos de manera automática. Por ejemplo, la base de datos usa el comando ROLLBACK para deshacer una transacción errónea. Cada uno cuenta con sus detalles de implementación, ventajas y casos de uso especializado

Selección y tipos de sistemas de gestión de bases de datos relacionales.

Existen varias ofertas en el mercado de sistemas, siempre lo ideal es escoger aquel que mejor se ajuste a la solución que se trata de implementar. Si bien, todos tienen la capacidad de almacenar y gestionar datos siguiendo los principios anteriormente vistos, hay detalles adicionales que se deben tener en cuenta al momento de elegir un programa en específico, por ejemplo:

- ¿Cuál es el volumen de datos que se requiere almacenar?
- ¿Qué tipos de datos se necesitan almacenar?
- ¿Cómo se accederá a los datos? (patrón de diseño de software)
- ¿Se necesitará en el futuro incrementar la capacidad de la base de datos?
- ¿Qué velocidad y rendimiento hace falta para la base de datos?
- ¿Qué características y funcionalidades se necesitan?
- ¿Debe ser compatible con algún sistema actualmente funcional?
- ¿Qué presupuesto se tiene para la implementación?
- ¿Qué medidas de seguridad tiene la base de datos?
- ¿Qué facilidad ofrece el sistema para la administración de la base de datos?

Un buen ejercicio inicial es, intentar responder estas preguntas y crear una lista de requerimientos, con lo que se puede facilitar la decisión. Algunas razones extra para elegir un programa en específico son:

MySQL:

Es un sistema de gestión de bases de datos de código abierto y gratuito. Entre sus ventajas está la facilidad de aprender, la escalabilidad y buen rendimiento que tienen y la comunidad de desarrolladores y usuarios con lo que se consigue buen soporte. Otra gran ventaja es que tiene programas que permiten la conexión entre muchos lenguajes de programación con el motor de bases de datos.

PostgreSQL

PostgreSQL es un sistema de código abierto y gratuito, estable y con buena seguridad en la información. Similar a MySQL es escalable, flexible y compatible con muchos lenguajes de programación. Adicionalmente tiene funciones avanzadas para la replicación y la recuperación ante desastres.

Microsoft SQL server

Dentro de las ventajas está la alta capacidad de hacer operaciones y la integración con los productos de Microsoft. Por eso se suele emplear con lenguajes de programación como C#. Tiene un sistema de seguridad informática sólido para repeler y resistir ataques de denegación de servicio y también cuenta con una comunidad grande de desarrolladores y usuarios.

Oracle database

Las bases de datos de Oracle han sido un estándar industrial debido a la confiabilidad de sus sistemas. Destaca por ser robusto y escalable, también porque es compatible con muchas soluciones empresariales para su despliegue en diferentes tipos de servidores. Desde Oracle proveen soporte para su uso, también cuenta con una comunidad de desarrolladores grande. De hecho, es una de las más populares junto con MySQL según la página



Herramientas de gestión de bases de datos



Junto con las bases de datos es común en etapas de desarrollo contar con sistemas que permiten la visualización rápida de las tablas, la creación y vista de relaciones, la creación de triggers y la gestión en general de las bases de datos. Por esto, muchos proveedores de bases de datos también ofrecen sistemas de gestión. Existen programas diseñados para una base de datos en específico como por ejemplo:

MySql Workbench para MySQL

SQL server management studio para SQL server

También existen sistemas que se pueden conectar a múltiples bases de datos como es el caso de DBeaver o Navicat.